JOURNAL OF COMPUTATIONAL PHYSICS 7, 239–254 (1971)

An Alternating Direction Implicit Method for Solving the Shallow Water Equations

BERTIL GUSTAFSSON†

National Center for Atmospheric Research,‡
Boulder, Colorado 80302

Received June 16, 1970

ABSTRACT

An alternating direction implicit difference scheme is applied to the shallow water equations on a β-plane. Unconditional stability of the scheme is proved for the linearized equations. For each time step a number of nonlinear systems of algebraic equations must be solved. Different iteration methods for doing this are discussed, and a quasi-Newton's method is developed, which can be used for arbitrarily large time steps. The scheme is tested numerically, and the different iteration methods are compared.

1. INTRODUCTION

When solving partial differential equations by explicit difference approximations, the time step is always restricted by a stability condition. In meteorological and oceanographic problems, one is often not interested in these short time steps because the discretization error in time is small compared to the discretization error in space. To avoid the stability condition, implicit schemes must be used, and some have been suggested and tested for this type of problem.

Most of them are "partly" implicit, so that there is still a stability condition, but a weaker one than for fully explicit schemes. The scheme considered in this paper is an alternating direction fully implicit scheme, and was stated in [7] (formulated differently) for linear initial boundary value problems.

It is applied to the shallow water equations, i.e., the primitive equations for an incompressible, inviscid fluid with a free surface. We will use the β-plane approximation on a rectangular domain. We will show that the method is unconditionally stable for the linearized equations.

† Present address: Department of Computer Sciences, University of Uppsala, Sweden.

‡ The National Center for Atmospheric Research is sponsored by the National Science Foundation.

581/7/2-4

Implicit schemes generally require more computation per time step than explicit ones. However, in many problems, as in meteorological applications, much computation which is independent of the particular type of difference scheme used, e.g., computing the forcing function, must be done for every time step. In these cases, the computing time required for obtaining the solution up to a certain time might depend more on the size of the time step than on the actual difference scheme chosen.

The scheme considered in this paper requires the solution of a number of nonlinear systems of algebraic equations. We will show that using a quasi-Newton method for this will yield a reasonable computing time for the propagation of the problem solution over a single time step, provided the solutions to the difference scheme equations are sufficiently smooth. Since the time step is determined only by considerations of accuracy and not of stability, the elapsed time required for obtaining a solution over a given time interval may be considerably reduced.

2. The Differential Equations and the Difference Scheme

Define the vector function $w = (u, v, \Phi)^T = w(x, y, t)$ (w^T denotes the transpose of a vector w), where u, v are the velocity components in the x- and y-direction respectively, and $\Phi = 2\sqrt{gh}$, where h is the depth of the fluid and g is the acceleration of gravity. Then the equations, obtained from [5, Eqs. (2.1–3)], have the form

$$\frac{\partial w}{\partial t} = A(w)\frac{\partial w}{\partial x} + B(w)\frac{\partial w}{\partial y} + C(y)w,$$

$$0 \leqslant x \leqslant L, \qquad 0 \leqslant y \leqslant D, \qquad 0 \leqslant t$$

where

$$A = -\begin{bmatrix} u & 0 & \Phi/2 \\ 0 & u & 0 \\ \Phi/2 & 0 & u \end{bmatrix}, \qquad B = -\begin{bmatrix} v & 0 & 0 \\ 0 & v & \Phi/2 \\ 0 & \Phi/2 & v \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & f & 0 \\ -f & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad f = \hat{f} + \beta(y - D/2), \quad \hat{f}, \beta \text{ const.}$$

We assume periodic solutions in the x-direction;

$$w(x, y, t) = w(x + L, y, t).$$

Then, with the boundary conditions

$$v(x, 0, t) = v(x, D, t) = 0,$$

and initial condition $w(x, y, 0) = \varphi(x, y)$, the energy

$$E = \frac{1}{2}\int_0^L \int_0^D \left(u^2 + v^2 + \frac{\Phi^2}{4}\right)\frac{\Phi^2}{4g}\, dx\, dy$$

is independent of the time. (Note that no boundary conditions are necessary for u and ϕ at $y = 0, D$.)

We define a grid vector function

$$w_{jk}^n = w(j\,\Delta x, k\,\Delta y, n\,\Delta t),$$

$$N_x\,\Delta x = L, \qquad N_y\,\Delta y = D;$$

the difference operators

$$D_{0x}w_{jk}^n = (w_{j+1,k}^n - w_{j-1,k}^n)/(2\Delta x),$$

$$D_{+x}w_{jk}^n = (w_{j+1,k}^n - w_{jk}^n)/\Delta x,$$

$$D_{-x}w_{jk}^n = (w_{jk}^n - w_{j-1,k}^n)/\Delta x,$$

(analogous for D_{0y}, D_{+y}, D_{-y}) and the operators

$$\begin{aligned} P_{jk}^n &= \frac{\Delta t}{2}\left(A(w_{jk}^n)\, D_{0x} + C_k^{(1)}\right) \\ Q_{jk}^n &= \frac{\Delta t}{2}\left(B(w_{jk}^n)\, D_k + C_k^{(2)}\right) \end{aligned} \tag{2.1}$$

with

$$D_k = \begin{cases} D_{0y} & \text{for} \quad k = 1, 2, \ldots, N_y - 1 \\ D_{+y} & \text{for} \quad k = 0 \\ D_{-y} & \text{for} \quad k = N_y, \end{cases} \tag{2.2}$$

$$C_k^{(1)} = \begin{bmatrix} 0 & 0 & 0 \\ -f_k & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad C_k^{(2)} = \begin{bmatrix} 0 & f_k & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Then the difference scheme is defined by

$$(I - P_{jk}^{n+1/2})\, w_{jk}^{n+1/2} = (I + Q_{jk}^{n})\, w_{jk}^{n}, \tag{2.3a}$$

$$(I - Q_{jk}^{n+1})\, w_{jk}^{n+1} = (I + P_{jk}^{n+1/2})\, w_{jk}^{n+1/2}. \tag{2.3b}$$

These equations do not apply to the v-component for $k = 0, N_y$, but we use the conditions

$$v_{j0} = v_{jN_y} = 0. \tag{2.4}$$

We assume throughout that $\lambda_x = \Delta t/\Delta x$, $\lambda_y = \Delta t/\Delta y$, λ_x, λ_y const. Adding the Eqs. (2.3a) and (2.3b) gives (cf. [9, p. 212])

$$w^{n+1} - w^n = 2P^{n+\frac{1}{2}}w^{n+\frac{1}{2}} + Q^n w^n + Q^{n+1}w^{n+1} = 2(P^{n+\frac{1}{2}} + Q^{n+\frac{1}{2}})\, w^{n+\frac{1}{2}} + \mathcal{O}(\Delta t^3),$$

where in the last equality we have assumed sufficiently smooth functions. Therefore, it is clear that the scheme is of second-order accuracy at all inner points, since centered difference operators are used there. At the boundaries $k = 0, N_y$, there is only first-order accuracy. However, we conjecture that the scheme has a convergence rate of second-order in the norm defined in [4, relation (2.7)].

3. Stability

We consider the linearized case obtained by setting $u_{jk}^n = \hat{u}_{jk}$, $v_{jk}^n = \hat{v}_{jk}$, $\Phi_{jk}^n = \hat{\Phi}_{jk}$ in the matrices A, B. In order to avoid technicalities, $\hat{u}$, $\hat{v}$, $\hat{\Phi}$ are assumed to be time independent.

We define a Hilbert space H by considering all vector functions satisfying $w_{jk} = w_{j+N_x,k}$; $v_{j0} = v_{jN_y} = 0$. The inner product of two vectors α, β and the norm are defined by

$$\begin{aligned} (\alpha, \beta) &= \Delta x\, \Delta y \sum_{j=1}^{N_x} \Big\{ \sum_{k=1}^{N_y - 1} \alpha_{jk}^{\mathrm{T}} \beta_{jk} + \frac{1}{2}(\alpha_{j0}^{\mathrm{T}}\beta_{j0} + \alpha_{jN_y}^{\mathrm{T}}\beta_{jN_y}) \Big\}, \\ \|\alpha\|^2 &= (\alpha, \alpha), \end{aligned} \tag{3.1}$$

where we have assumed real-valued vector functions, which is no restriction in this case.

From well-known identities for difference operators, it is clear that if A is uniformly bounded and uniformly Lipschitz continuous in x and y, then

$$\Delta t\, AD_{0x} = \frac{\Delta t}{2}(AD_{0x} + D_{0x}A) + \mathcal{O}(\Delta t). \tag{3.2}$$

(We will use throughout this paper the notation $\mathcal{O}(\Delta t)$ for operators E with $\| E \| \leqslant \text{const} \cdot \Delta t$, for vectors α with $\| \alpha \| \leqslant \text{const} \cdot \Delta t$, and for scalar functions $\varphi(\Delta t)$ with $| \varphi(\Delta t)| \leqslant \text{const} \cdot \Delta t$.) Analogous formulas are valid for D_{0y}, $D_{\pm y}$, and by defining

$$\begin{aligned} \tilde{P}_{jk} &= \frac{\Delta t}{4}(A_{jk}D_{0x} + D_{0x}A_{jk}), \\ \tilde{Q}_{jk} &= \frac{\Delta t}{4}(B_{jk}D_k + D_kB_{jk}), \end{aligned} \tag{3.3}$$

D_k defined by Eq. (2.2), it is clear that

$$\begin{aligned} \tilde{P}_{jk} &= P_{jk} + \mathcal{O}(\Delta t), \\ \tilde{Q}_{jk} &= Q_{jk} + \mathcal{O}(\Delta t). \end{aligned} \tag{3.4}$$

We will first show that the scheme (2.3) with $\tilde{P}_{jk}$, $\tilde{Q}_{jk}$ substituted for P_{jk}, Q_{jk} is stable. In [7] it is shown that if the relations

$$(w, \tilde{P}w) = 0 \tag{3.5a}$$

$$(w, \tilde{Q}w) = 0 \tag{3.5b}$$

are valid, the scheme is stable. The proof of this can be simply written as

$$\begin{aligned} \| w^{n+1} \|^2 + \| \tilde{Q}w^{n+1} \|^2 &= \| w^{n+1} \|^2 + \| \tilde{Q}w^{n+1} \|^2 - 2(w^{n+1}, \tilde{Q}w^{n+1}) \\ &= \|(I - \tilde{Q})\, w^{n+1} \|^2 = \|(I + \tilde{P})\, w^{n+\frac{1}{2}} \|^2 = \|(I - \tilde{P})\, w^{n+\frac{1}{2}} \|^2 \\ &= \|(I + \tilde{Q})\, w^n \|^2 = \| w^n \|^2 + \| \tilde{Q}w^n \|^2. \end{aligned}$$

$\tilde{Q}$ is a bounded operator, so this equality is equivalent to stability. Equation (3.5a) is immediately clear, as A is symmetric and the operator D_{0x} is antisymmetric in H:

$$\begin{aligned} (w, (AD_{0x} + D_{0x}A)w) &= -(D_{0x}Aw, w) - (AD_{0x}w, w) \\ &= -(w, (AD_{0x} + D_{0x}A)w) = 0. \end{aligned}$$

To show Eq. (3.5b) we can without restriction assume $\hat{v}_{j1} = \hat{v}_{j,N_y-1} = 0$; the assumption of uniform Lipschitz-continuity means that the error introduced can be included in the $\mathcal{O}(\Delta t)$ term in Eqs. (3.4). To make the proof of Eq. (3.5b) more readable, we will neglect the boundary $k = N_y$, assuming for instance that $w_{jN_y} = 0$. Define

$$(\alpha, \beta)' = \Delta x\, \Delta y \sum_{j=1}^{N_x} \sum_{k=1}^{N_y-1} \alpha_{jk}^{\mathrm{T}} \beta_{jk}.$$

Then we have, with $c_{j0} = \lambda_y \hat{\Phi}_{j0}/4$,

$$4(w, \tilde{Q}w) = \Delta t(w, (BD_{0y} + D_{0y}B)w)' + \Delta x\, \Delta y \sum_{j=1}^{N_x} (-\Phi_{j0}c_{j0}v_{j1} - \Phi_{j0}c_{j1}v_{j1})$$

$$= -\Delta t(D_{0y}Bw, w)' + \Delta x\, \Delta y \sum_{j=1}^{N_x} (v_{j1}c_{j1}\Phi_{j0} + c_{j0}\Phi_{j0}v_{j1})$$

$$-\Delta t(BD_{0y}w, w)' + \Delta x\, \Delta y \sum_{j=1}^{N_x} (v_{j1}c_{j0}\Phi_{j0} + c_{j1}\Phi_{j0}v_{j1})$$

$$+\Delta x\, \Delta y \sum_{j=1}^{N_x} (-\Phi_{j0}c_{j0}v_{j1} - \Phi_{j0}c_{j1}v_{j1})$$

$$= -4(w, \tilde{Q}w) = 0.$$

To show stability for the original scheme (2.3), we first note that it can be written (omitting subscripts)

$$(I - P)(I - Q)\, w^{n+1} = (I + P)(I + Q)\, w^n. \tag{3.6}$$

The relations (3.5a, b) show that the inverse operators $(I \pm \tilde{P})^{-1}$, $(I \pm \tilde{Q})^{-1}$ exist and are bounded, therefore Eq. (3.6) can be written

$$(I - \tilde{P})(I - \tilde{Q})(I + \mathcal{O}(\Delta t))\, w^{n+1} = (I + \tilde{P})(I + \tilde{Q})(I + \mathcal{O}(\Delta t))\, w^n,$$

and we have

$$w^{n+1} = (I + \mathcal{O}(\Delta t))^{-1} (I - \tilde{Q})^{-1} (I - \tilde{P})^{-1} (I + \tilde{P})(I + \tilde{Q})(I + \mathcal{O}(\Delta t))\, w^n.$$

Accordingly, for small Δt we have, with $\| w \|_{\tilde{Q}} = \| w \|^2 + \| \tilde{Q}w \|^2$,

$$\| w^{n+1} \|_{\tilde{Q}} \leqslant (1 + \mathcal{O}(\Delta t)) \| w^n \|_{\tilde{Q}},$$

and

$$\| w^n \|_{\tilde{Q}} \leqslant (1 + \mathcal{O}(\Delta t))^n \| w^0 \|_{\tilde{Q}} \leqslant K_1 e^{n\Delta t k_2} \| w^0 \|_{\tilde{Q}}.$$

So, on any finite time interval $(0, T)$, we finally obtain

$$\| w^n \| \leqslant K_3 \| w^0 \|,$$

and the stability is proved.

4. Solving the Systems of Algebraic Equations

For each time step of the scheme, a number of nonlinear systems of algebraic equations have to be solved. If the systems are written in the form $w = r(w)$, the simple iteration technique

$$w^{(m+1)} = r(w^{(m)}), \qquad m = 0, 1, \ldots, p \tag{4.1}$$

(hereafter called GIp) can be used. This method has the advantage of being fast in the sense that each iteration step can be carried out using a comparatively small number of arithmetic operations, and is easy to program. However, the convergence criterion imposes an upper limit on λ_x, λ_y, which in this case can be shown to be approximately four times as large as the Courant–Friedrich–Lewy limit for explicit schemes. Furthermore, the convergence may be slow, particularly if λ_x, λ_y are near the convergence limit and the solution to the differential equation varies rapidly with time.

Also we have to take into consideration the results found by Gary [2] concerning the iterative solution of the system of equations arising from the Crank–Nicholson scheme for a linear equation. These results can easily be transferred to our scheme, and show that the number of iterations in each half time step has to be chosen from the sequence 3, 4, 7, 8, 11, 12,..., in order to avoid instability for the linear case. With these p-values, a certain order of dissipation is obtained. The method (4.1) was tested, and the results are discussed in Section 5.

We will now describe a quasi-Newton's method, where we are able to do a rigorous analysis of the order of accuracy.

No more than two variables are coupled to each other on the leftsides of Eqs. (2.3), and we will here describe how to solve for $(u^{n+\frac{1}{2}}, \Phi^{n+\frac{1}{2}})$ from the first and third equation of (2.3a). The equations are written in the form

$$g(\alpha) = 0, \tag{4.2}$$

where

$$\alpha = (u_1, \Phi_1, u_2, \ldots, \Phi_{N_x})^{\mathrm{T}}$$

(the n- and k-indices are omitted). The original Newton's method, described e.g. in [6, Chap. 3], is given by

$$\alpha^{(m+1)} = \alpha^{(m)} - J^{-1}(\alpha^{(m)})\, g(\alpha^{(m)}), \tag{4.3}$$

where the superscript denotes iteration index, and J is the Jacobian

$$J = \partial(g, \alpha)$$

with the form

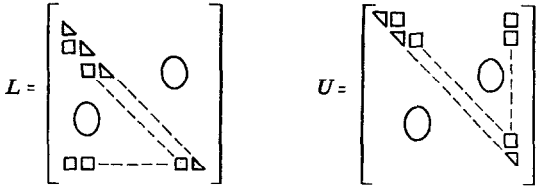
$$J = \begin{bmatrix} D_1 & H_1 & & & & -H_1 \\ -H_2 & D_2 & H_2 & & & 0 \\ & -H_3 & D_3 & H_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \\ & 0 & & & \ddots & H_{N_x-1} \\ H_{N_x} & & & & -H_{N_x} & D_{N_x} \end{bmatrix} \tag{4.4}$$

where

$$H_j = \frac{\lambda_x}{8}\begin{bmatrix} 2u_j & \Phi_j \\ \Phi_j & 2u_j \end{bmatrix},$$

$$D_j = \begin{bmatrix} 1 + \frac{\lambda_x}{4}(u_{j+1} - u_{j-1}) & \frac{\lambda_x}{8}(\Phi_{j+1} - \Phi_{j-1}) \\ \frac{\lambda_x}{4}(\Phi_{j+1} - \Phi_{j-1}) & 1 + \frac{\lambda_x}{8}(u_{j+1} - u_{j-1}) \end{bmatrix}.$$

In order to solve for $J^{-1}g$, J is decomposed into $J = LU$, (see e.g. [6, Sec. 2.3.3]) where L, U have the forms



(the squares and triangles mean (2×2)-matrices).

$J^{-1}g$ is then computed by backsubstitution, i.e. z is first solved for from $Lz = g$, and $J^{-1}g$ from $U(J^{-1}g) = z$. The quasi-Newton's method now means that the LU decomposition is done only every M-th time step, where M is a fixed integer. Since the backsubstitution is a fast operation, the scheme will be efficient, provided the number of iterations is small.

To check the order of accuracy we assume throughout that the solutions are sufficiently smooth. If our starting approximation $\alpha^{(0)}$ is taken to be the values computed at the previous time step, we then have

$$\| \xi - \alpha^{(0)} \| = \mathcal{O}(\Delta t), \tag{4.5}$$

where $g(\xi) = 0$. (In this section $\| \ \|$ implies summation only over j.)

The iteration formula is

$$\alpha^{(m+1)} = \alpha^{(m)} - \hat{J}^{-1}(\alpha^{(m)})\, g(\alpha^{(m)}), \tag{4.6}$$

where

$$\hat{J} = J(\alpha^{(0)}) + \mathcal{O}(\Delta t). \tag{4.7}$$

Equation (4.7) is true if M is any fixed number and if we consider the limit process $\Delta t \to 0$. However, from a practical point of view, it is useful only if M is a relatively small number. A Taylor expansion gives, when taking Eq. (4.5) into account,

$$0 = g(\xi) = g(\alpha^{(0)}) + J(\alpha^{(0)})(\xi - \alpha^{(0)}) + \mathcal{O}(\Delta t^2). \tag{4.8}$$

Assuming that $\hat{J}^{-1}$ exists and is uniformly bounded when $\Delta t \to 0$, we can multiply Eq. (4.8) from the left by $\hat{J}^{-1}$, and obtain, taking Eqs. (4.5), (4.6), (4.7) into account,

$$0 = \hat{J}^{-1}g(\alpha^{(0)}) + \xi - \alpha^{(0)} + \hat{J}^{-1}\mathcal{O}(\Delta t^2) = \alpha^{(0)} - \alpha^{(1)} + \xi - \alpha^{(0)} + \mathcal{O}(\Delta t^2).$$

Accordingly,

$$\| \xi - \alpha^{(1)} \| = \mathcal{O}(\Delta t^2),$$

and in general

$$\| \xi - \alpha^{(m)} \| = \mathcal{O}(\Delta t^{m+1}). \tag{4.9}$$

For Eq. (4.9) to be valid, we must show

$$\| \hat{J}^{-1} \| \leqslant K, \qquad K \text{ independent of } \Delta t, \quad \Delta t \to 0.$$

The representation (4.4) shows that

$$\hat{J} = I + E + \mathcal{O}(\Delta t),$$

where E is an antisymmetric matrix. The eigenvalues κ_j of $I + E$ have the form

$$\kappa_j = 1 + i\gamma_j, \qquad \gamma_j \text{ real},$$

with

$$| \kappa_j |^2 = 1 + \gamma_j^2,$$

and we have for $\Delta t \leqslant$ some const

$$\| \hat{J}^{-1} \| \leqslant \|(I + E)^{-1}\| + \mathcal{O}(\Delta t) = (\min_j | \kappa_j |)^{-1} + \mathcal{O}(\Delta t) \leqslant 1.$$

This shows that Eq. (4.9) is valid, and also that the inversion of $\hat{J}$ is a very well conditioned problem.

When $u^{n+\frac{1}{2}}$, $\Phi^{n+\frac{1}{2}}$ are known, $v^{n+\frac{1}{2}}$ can be determined in the same way, except that D_j, H_j in Eq. (4.4) are now scalars. We solve for w^{n+1} in exactly the same way, except that u and v are interchanged and the "extra lines and columns" in L and U respectively do not appear because the boundary conditions are not periodic in the y-direction.

Equation (4.9) shows that two iterations give sufficient accuracy, and we call this method QN2. However, if we obtain $\alpha^{(0)}$ by doing linear extrapolation in time using the solutions at the two latest known times levels, we have $\| \xi - \alpha^{(0)} \| = \mathcal{O}(\Delta t^2)$. It is then immediately clear that $\| \xi - \alpha^{(1)} \| = \mathcal{O}(\Delta t^3)$, which means that one iteration is enough. (This method will be called QNEX1.)

As the exact solution of the difference scheme is not obtained in either case, there could be a slight growth in the solution. To investigate this growth for QN2, we consider the amplification factor μ for the scalar case and only one space dimension, i.e., for the equation

$$(1 - aD_0)\, w^{n+1} = (1 + aD_0)\, w^n. \tag{4.10}$$

After Fourier transformation, $\hat{J}^{-1}$ corresponds to $(1 - i(\lambda/2)\hat{a} \sin \omega)^{-1}$, where $| a - \hat{a} | = \Delta a = \mathcal{O}(\Delta t)$. After two iterations we have

$$\mu = \left(1 - i\hat{b} + 2ib - (1 - ib)\left(1 + \frac{2ib}{1 - i\hat{b}}\right) + 1 + ib\right)\Big/(1 - i\hat{b}),$$

where

$$b = \frac{\lambda}{2} a \sin \omega, \qquad \hat{b} = \frac{\lambda}{2} \hat{a} \sin \omega.$$

After some calculation we obtain

$$| \mu |^2 = \frac{1 + 2\hat{b}^2 + \hat{b}^4 - 4\hat{b}^2(b - \hat{b})^2 + 4(b - \hat{b})^4}{1 + 2\hat{b}^2 + \hat{b}^4} \leqslant 1, \tag{4.11}$$

where we have assumed $| \Delta a | \leqslant | \hat{a} |$. Further, Eq. (4.11) shows that even if $| \hat{a} | < | \Delta a |$ the growth of the solution is very small, for in that case $| \mu | \leqslant 1 + \mathcal{O}(\Delta t^4)$.

The corresponding investigation for the QNEX1 method was not carried out, because it has to be done for two space dimensions. However, we know that $| \mu | = 1 + \mathcal{O}(\Delta t)$ in this case. If the solution is wanted over a very long time interval, this growth could be eliminated by adding a dissipation term without changing the accuracy of the scheme.

To determine the efficiency of the scheme as applied to the equations treated in this paper, an operation count is useful.

For all three methods, the number of operations per full time step is KN_xN_y, where K depends on the method.

We have for our three methods,

$$K_{\mathrm{GI}_p} = 38 + p \cdot 22, \tag{4.12a}$$

$$K_{\mathrm{QNEX1}} = 115 + 152/M, \tag{4.12b}$$

$$K_{\mathrm{QN2}} = 210 + 152/M, \tag{4.12c}$$

where p is the number of iterations for the GI method, and M is the number of time steps between the LU decompositions. Therefore, with the choice $p = 3$ and $M = 12$ (used in numerical experiments), we have

$$K_{\mathrm{GI3}} = 104,$$

$$K_{\mathrm{QNEX1}} = 128,$$

$$K_{\mathrm{QN2}} = 223.$$

As a comparison, we have for the simplest formulation of the leapfrog scheme without any averaging in space $K_{\mathrm{LEAPFROG}} = 42$.

5. Numerical Results

The purpose of this section is primarily to investigate the practical applicability of the arguments in Section 4.

The program was run on the CDC 6600 computer at the Computing Facility of the National Center for Atmospheric Research. We always used the initial function used by Grammeltvedt [3, initial condition I] and Williamson [10], i.e., the height field

$$h(x, y) = H_0 + H_1 \tanh\left(\frac{9(D/2 - y)}{2D}\right) + H_2 \operatorname{sech}^2\left(\frac{9(D/2 - y)}{D}\right) \sin\left(\frac{2\pi x}{L}\right)$$

and geostrophic velocity fields, i.e. $u = -(g/f)\,\partial h/\partial y$, $v = (g/f)\,\partial h/\partial x$. Constants used were $L = 4400$ km, $D = 6000$ km, $f = 10^{-4}$ sec^{-1}, $\beta = 1.5 \times 10^{-11}$ sec^{-1} m^{-1}, $g = 10$ m sec^{-2}, $H_0 = 2000$ m, $H_1 = 220$ m, $H_2 = 133$ m.

The scheme was run initially with the resolution $\Delta x = \Delta y = 200$ km, and $\Delta t = 1800$ sec, which means that λ was approximately 3 times larger than the C–F–L limit for explicit schemes for this problem. Figure 1 shows the height field as

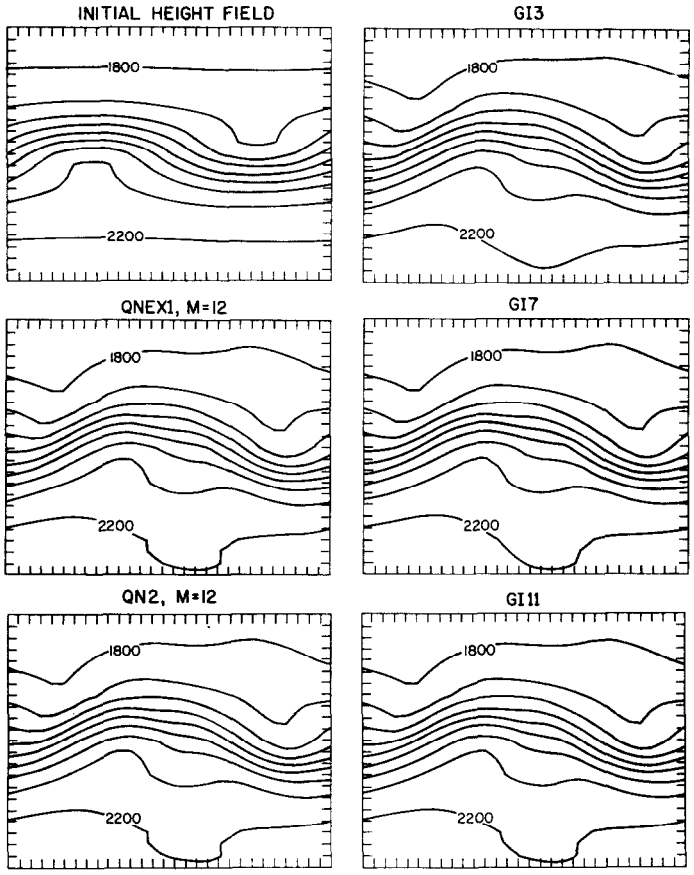


FIG. 1. The initial height field, and the height field after two days for the different methods; $\Delta t = 1800$ sec.

a result of QNEX1, QN2 ($M = 12$), and GI3, GI7, GI11 after 2 days. We chose this point of time because the true solution of the scheme, i.e. the one obtained after a large number of iterations, has a little wrinkle on the 2200 contour, which is a good test of the convergence properties of the iteration methods. A visual inspection shows that QNEX1 and QN2 give equal results, while the GI method needs 11 iterations to yield a comparable solution. However, the smooth part of the solution is accurate after three iterations, as was true in all the experiments.

Figure 2 shows QNEX1, QN2, and QN3 run with $\Delta t = 3600$ sec, $M = 6$ (λ then exceeds the convergence limit for the GI method). Here we can discern a small discrepancy between QNEX1 and QN2 in the "eastern" parts of the 2200 contours.

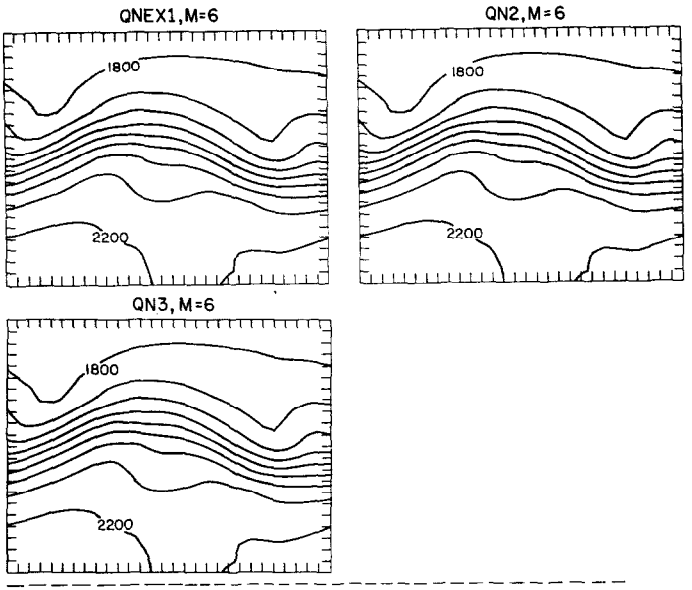


FIG. 2. The height field after two days for the QN methods; $M = 6$, $\Delta t = 3600$ sec.

The error between the approximate and the true solution of the scheme is shown in Table I for the different methods. The true solution is represented by w_{QN3}, and ϵ is defined by $\epsilon_I = w_I - w_{QN3}$, where I stands for any of the iteration methods. The norm used is the one defined by Eqs. (3.1).

TABLE I

$\| \epsilon_I \| / \| w_{QN3} \|$, $t = 2$ days

Method	$\Delta t = 1800$ sec	$\Delta t = 3600$ sec
GI3	1.3×10^{-3}	—
GI7	2.9×10^{-4}	—
GI11	7.5×10^{-5}	—
QNEX1	5.6×10^{-5}	1.4×10^{-4}
QN2	6.4×10^{-7}	4.9×10^{-6}

We believe that the error for both QN methods is much less than the difference between the true solution to the difference scheme and the solution to the differential equation.

Some long runs were also made. The solutions always "blew up" after approximately 18 days, regardless of the iteration technique, which means that the "explosion" was caused by nonlinear instabilities in the scheme. We do not consider this property of the scheme as a significant disadvantage, since in long time integrations of meteorological and oceanographic problems there is always some kind of dissipation in the system.

As noted in Section 4, the GIp-iteration method automatically enters dissipation into the scheme for $p = 3, 4, 7, 8, 11, \ldots$. For $p = 3$, which yields dissipation of fourth order, the "explosion" mentioned above does not occur (see Fig. 3).

A smoothing of the solutions, using the QN method, can be achieved in various ways. We added the dissipation term $\epsilon \Delta t^3 D_{+y} D_{-y} w_{jk}^n$ to the rightside of Eq. (2.3a), and the term $\epsilon \Delta t^3 D_{+x} D_{-x} w_{jk}^{n+\frac{1}{2}}$ to the rightside of Eq. (2.3b). The result, with $\epsilon = 0.015$, is shown in Fig. 3, together with the results of the other methods. ($\Delta x = \Delta y = 5 \times 10^5$ m always.) We believe, though, that a better way to handle the nonlinear instabilities is to use a nonlinear eddy viscosity term as described in [1] and [8].

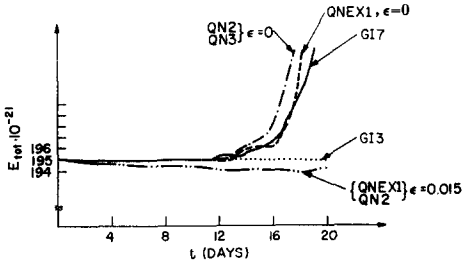


FIG. 3. Total energy for the different methods, $\Delta x = \Delta y = 5 \times 10^5$, $\Delta t = 3600$ sec. $M = 12$ for QN methods; $\Delta t = 1800$ for GI methods.

The following table shows the run time in seconds per full time step with $\Delta x = \Delta y = 2 \times 10^5$ m for the different iteration methods used.

TABLE II

GI3	0.16
QNEX1, $M = 12$	0.43
QNEX1, $M = 6$	0.49
QN2, $M = 12$	0.74
QN2, $M = 6$	0.80

6. CONCLUSIONS

The scheme described in this paper is unconditionally stable for the linearized case; hence only the desired discretization error governs the selection of the time-step when the quasi-Newton method is used to solve the algebraic equations. This advantage becomes more evident when the equations are applied on the globe: we avoid the severe restriction on the time step for conditionally stable schemes arising from the converging grid near the poles.

Disadvantages of the QN methods (with $M > 1$) are that extra storage is required for the L, U matrices and for the function values at the previous time level for QNEX1, and that the programming becomes comparatively complicated, as is reflected in Table II where the run times do not compare to the operation count in Section 4.

For the initial function and the λ values used in the numerical experiments, the error in the solution of the difference equations by QNEX1 appears to be less than the discretization error; hence it is probably satisfactory in most cases.

If we are interested in moderate λ values, the GI method can be used for solving the algebraic equations. It is simple to program and is fast if the number of iterations can be kept small. However, a rigorous analysis of the accuracy is more difficult in this case, and we cannot expect the high wave number components to be well represented. Nevertheless, this might be an advantage if one is interested in long time integrations, since dissipation is automatically built into the scheme.

ACKNOWLEDGMENTS

My thanks are due to Professor Heinz-Otto Kreiss for a number of stimulating discussions and to Mr. Jack Miller of NCAR, who did the programming.

REFERENCES

1. W. P. CROWLEY, A numerical model for viscous, free surface, barotropic wind driven ocean circulations, *J. Comput. Phys.* **5** (1970), 139–168.
2. J. GARY, On certain finite difference schemes for hyperbolic systems, *Math. Comp.* **18** (1964), 1–18.
3. A. GRAMMELTVEDT, A survey of finie-difference schemes for the primitive equations for a barotropic fluid, *Mon. Weather Rev.* **97** (1969), 384–404.
4. B. GUSTAFSSON, H.-O. KREISS, AND A. SUNDSTRÖM, Difference approximations to mixed initial boundary value problems. II. *Math. Comp.*, to appear.
5. D. HOUGHTON, A. KASAHARA, AND W. WASHINGTON, Long-term integration of the barotropic equations by the Lax–Wendroff method, *Mon. Weather Rev.* **94** (1966), 141–150.
6. E. ISACSON AND H. B. KELLER, "Analysis of Numerical Methods," John Wiley and Sons, Inc., New York, 1966.

7. H.-O. KREISS, AND O. B. WIDLUND, Difference approximations for initial value problems for partial differential equations, *in* "Proceedings of the Summer School for Mathematics and Physics in Münich, 1966," (M. Kruskal, Ed.), Springer-Verlag, New York, to appear.
8. C. E. LEITH, Two dimensional eddy viscosity coefficients, *in* "Properties of matter under unusual conditions," (H. Mark, S. Fernbach, Eds.), pp. 267–271, Interscience Publishers, New York, 1969.
9. R. D. RICHTMYER AND K. W. MORTON, "Difference Methods for Initial-Value Problems," 2nd ed., Interscience, New York, 1967.
10. D. WILLIAMSON, Numerical integration of fluid flow over triangular grids, *Mon. Weather Rev.* **97** (1969), 885–895.